Contents

WEBTRENDS

Collecting Web Traffic Data Across the Enterprise With Data Collection Server

White Paper
July 3, 2002

Over the last decade economies have become global and many organizations maintain offices around the world, relying heavily on the Internet and corporate networks for communication. Many of these geographically dispersed sites maintain their own web sites, using local web hosting services—as a matter of convenience or because they may have region-specific objectives that can be carried out more easily by locally-maintained and hosted sites.

This creates difficulties when trying to analyze the web data from all the sites as a whole, since each site generates its own web server log.

Many other situations also create obstacles for analyzing and reporting on web site traffic. These include web site hosting services that do not provide web server logs to their clients, and caching servers that store web pages in short term cache memory for fast page reloads when the web visitor decides to revisit the page.

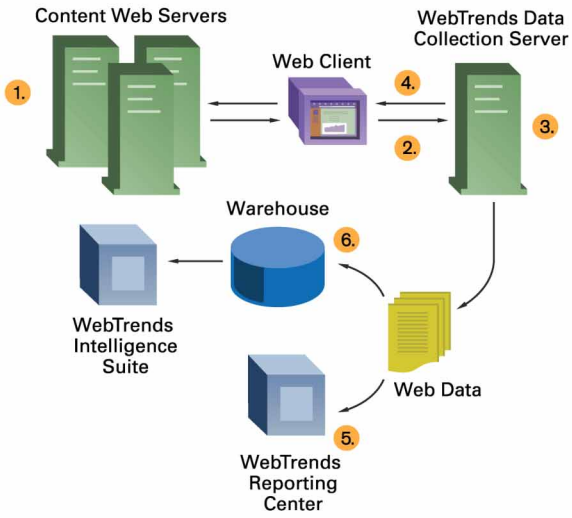
The common element is the need to obtain web analytics without the reliance on web server logs generally used in this process. This white paper seeks to:

- Explain an alternative mechanism of data collection that eliminates the need for log files and introduce the WebTrends Data Collection Server, which uses this approach.
- Address pre-implementation considerations for the Data Collection Server.
- Discuss differences between standard web server log files and the Data Collection Server log files.

An Alternative Approach to Data Collection

To address the issues presented by multi-national companies, businesses with multiple web sites, web sites using caching servers, or hosted sites with unavailable web log files, NetIQ has developed the WebTrends Data Collection Server (DCS). Rather than relying on web server log files, this solution collects web traffic data by including JavaScript in web content sent to client machines. The script directs hit information to DCS, which stores the information in a format that is fully compatible with WebTrends Reporting Center and WebTrends Intelligence Suite. While reports can be run against this data source, the information can also be stored in the WebTrends Warehouse for further integration with other data sources or for more in-depth ad-hoc analysis and Online Analytical Processing (OLAP).

This diagram shows how DCS works with WebTrends solutions to collect web traffic data:



1. Web server delivers web content containing JavaScript to the client machine.
2. JavaScript directs hits to WebTrends® Data Collection Server™ (DCS).
3. DCS validates returning visitor or generates cookie (optional) and logs the hit.
4. If client is new, DCS delivers a cookie (optional) to the client machine; if a returning client, the cookie is read to properly sessionize visitors.
5. WebTrends Reporting Center™ analyzes resulting data.
6. WebTrends Warehouse Builder™ produces a warehouse that is compatible with all other WebTrends enterprise level products.

Understanding the Basics of JavaScript

JavaScript can be imbedded in web pages capable of executing the code, for example, .html, .htm and .asp web pages. When a web visitor requests the page from the server, the browser on the client machine displays the page according to the page code, running the JavaScript code when it encounters the code on the page and providing the functionality the code requests.

For example, if you want to display a note saying "Last Updated: " with the actual date the document was last updated, you would place the following JavaScript on the page:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write("Last Updated: ");
document.write(document.lastModified);
// -->
</SCRIPT>
```

You would insert this snippet of JavaScript into the page's source code at the point in the page that you want it to be displayed. Because the code is within <!-- ... -->, it does not cause problems for clients using browsers that don't support JavaScript, which industry information indicates is less than 1% of visitors.

The above code instructs the client's browser to insert the words "Last Updated: ", followed by the date the document was last modified. The property "document.lastModified" pulls that information from the document's file properties, and places it in the displayed Web page as requested by the "document.write" method.

Understanding the Data Collection Server JavaScript Tag

The following JavaScript tag is used by the Data Collection Server to collect basic web hit data:

```
<!-- START OF Data Collection Server TAG -->
<!-- Copyright 2002 NetIQ Corporation -->
<!-- V2.0 -->
<SCRIPT LANGUAGE="JavaScript">
<!--
function dcs_2_0(dcs_URI,dcs_QRY,dcs_EXT)
{
       var dCurrent = new Date();
       var P = "";
       P+="http"+(window.location.protocol.indexOf('https:')==0?'s':'')+"://localhost/dcs.
gif?";
       P+="dcsuri="+escape(dcs_URI);
       P+="&dcsqry="+escape(dcs_QRY);
       if ((window.document.referrer != "") && (window.document.referrer != "-"))
       {
              if (!(navigator.appName == "Microsoft Internet Explorer" &&
parseInt(navigator.appVersion) < 4) )
          {
              P +="&dcsref="+escape(window.document.referrer);
          }
       }
       P+=dcs_EXT;
       //P+="&dcssip=yourdomain";  //For Cross domain tracking, replace 'yourdomain' and
remove the leading '//'.
       //P+="&dcsp3p=yourp3pheader";  //To issue P3P header, replace 'yourp3pheader' and
remove the leading '//'.
```

Creates a value for the variable P that consists of the URI plus the query string to show exactly the page that the visitor visited.

```
       //P+="&dcscfg=yourcfg";  //To configure DCS, replace 'yourcfg' and remove the
leading '//'.
       P+="&dcsdat="+escape(dCurrent.getTime());
       document.write('<IMG BORDER="0" NAME="DCSIMG" WIDTH="1" HEIGHT="1" SRC="'+P+'">');
}
//-->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<!--
function dcsExtend(N,V)
{
       dcsEXT+=”&”+N+”=”+escape(V);
}
function dcsMeta()
{
       var F=false;
       var myDocumentElements;
       if (document.all)
       {
              F = true;
              myDocumentElements=document.all.tags(“meta”);
       }
       if (!F && document.documentElement)
       {
              F = true;
              myDocumentElements=document.get.ElementsByTagName(“meta”);
       }
              if (F)
       {
              for (var I=1; I<=myDocumentElements.length;I++)
              {
                     myMeta=myDocumentElements.item(I-1);
                     if (myMeta.name.indexOf(‘WT.’)==0)
                            dcsExtend(myMeta.name,myMeta.content);
              }
       }
}
var dcsURI=window.location.pathname;
var dcsQRY=window.location.search;
var dcsEXT=””;
dcsMeta();
dcs_2_0(dcsURI,dcsQRY,dcsEXT);
//-->
</SCRIPT>
<NOSCRIPT>
<IMG BORDER="0" NAME="DCSIMG" WIDTH="1" HEIGHT ="1" SRC="http://localhost/njs.gif?dcsuri=/nojavascript">
</NOSCRIPT>
<!-- END OF Data Collection Server TAG -->
```

Requests the image dcs.gif from the Data Collection Server, and appends the now-defined variable “P” to the request. DCS then translates this information into a standard log file format.

Joins together WebTrends <META> tags for use as query parameters.

If the browser does not support JavaScript, the DCS sends a no-JavaScript image back (njs.gif) and assign the URI a value of “nojavascript.” Thus, page views and visitors are accurately counted whether JavaScript is enabled or disabled.

In the above example, if the client's web browser is JavaScript-enabled, when he opens a web page containing the JavaScript tag, the Browser makes an image request to the Data Collection Server and attaches the value of the variable P to that request. The value contains a string with the URL, the query string, the referring page, and the date and time of the visit. The Data Collection Server captures that variable value, which at this point contains basic hit information, and generates a Data Collection Server log file. The hit information in this log file can be used immediately with WebTrends Reporting Center, or may be stored in the WebTrends warehouse for more in-depth analysis and reporting using WebTrends Intelligence Suite solutions, such as WebTrends Report Designer and OLAP Manager.

If the client's browser is not JavaScript-enabled, the hit and visit are still tallied, but the more detailed information (e.g., page name, URL, and referrer) is not defined.

Setting Up the Data Collection Server

In this section, learn how to set up your environment to best use the Data Collection Server.

Operating Environment

As a cross-platform solution, the Data Collection Server can accommodate a variety of operating environments, including:

Operating Systems

- Microsoft® Windows® NT® 4.0, Service Pack 3 and Windows 2000/XP
- Sun™ Solaris™ 2.7 and 2.8
- Red Hat® Linux® 6.2 or later

Hardware Platforms

X86: Dual 1 GHz processors (used with Windows NT and Red Hat Linux)

Solaris: Dual 400 MHz UltraSPARC-II processors

- 1 GB free memory
- Enough disk space to store the expected contents of your log files (SCSI recommended)

For larger sites and sites that receive high levels of web traffic, it may be necessary to increase the number of processors, the speed of the processors, the amount of memory and the network bandwidth of the DCS host. You can also provide virtually unlimited scalability of DCS or plan for redundancy by employing multiple machines with load-balancing techniques.

Determining the Number of Data Collection Servers Required

Because DCS must be contacted with each page view of your site, it must be running on hardware adequate to handle the incoming requests. In order to properly assess what is required, it is important to have an understanding of both the total number of page views your site gets per day, and also the peak connections per second that must be managed.

The following combinations of web servers and operating systems have been tested and deliver very similar performance in terms of handing overall traffic and peak load:

- IIS on Windows NT/2000
- Apache on Linux
- IPlanet on Solaris

Each of these combinations can easily manage sites that receive up to 50 million page views per day provided that peak volume not exceed 2000 connections per second. If site traffic exceeds these levels, or if 100% fail-over redundancy is required (recommended), a second DCS must be added. DCS servers can be load balanced to provide unlimited scalability and redundancy.

Network Considerations

DCS is a specialized web server that receives HTTP requests from web clients, processes these requests, and appropriately responds to the web clients. The connections established between the clients and the server use TCP/IP protocol. This means that to make the required connection, the web server must be listening on a predetermined port associated with a known IP address, and the client must have knowledge of the IP address and connectivity to the web server. This is accomplished by installing and configuring the Data Collection Server on the network just as you would any other web server, and then modifying a portion of the DCS JavaScript tag to establish the location of that installed Data Collection Server.

Before installing DCS, consider the various issues related to the configuration of your network that enable the TCP/IP communication between the web server and web clients.

Security Issues

Because external clients can make requests to the DCS server, security issues should be considered prior to installation. DCS is simply a specialized web server, and for this reason, the security issues related to DCS are common to all standard web servers—unauthorized access to confidential data, tampering with web site behavior and functionality, data corruption and complete denial of service (DOS) on your web site.

To combat security issues, you could take the approach of initially configuring the DCS server in a "deny all" mode, which begins with all services disabled, and services necessary for the operation of the DCS server would subsequently be enabled when they were deemed necessary. Specifically, services (daemons) such as Telnet, mail, and finger should not be enabled. For the DCS server, the only TCP/IP link required to the Internet would be port 80.

In addition, you may choose to use a firewall, a security solution that is often employed to protect web servers and/or internal corporate networks. In using a firewall, you will need to determine the location of your DCS server relative to any firewall in your network, just as you have with your other web servers If you place the DCS server on the outside of the firewall, it will be more susceptible to malicious attacks. In the event of a break-in, however, the attacker will have only breached the boundaries of the DCS server and not the entire corporate network. It is highly recommended that you research and address security concerns before exposing the DCS server to external users.

P3P

The Platform for Privacy Preferences The Platform for Privacy Preferences Project (P3P), developed by the World Wide Web Consortium, is emerging as an industry standard providing a simple, automated way for users to gain more control over the use of personal information on Web sites they visit. At its most basic level, P3P is a standardized set of multiple-choice questions, covering all the major aspects of a Web site's privacy policies. Taken together, they present a clear snapshot of how a site handles personal information about its users. P3P-enabled Web sites make this information available in a standard, machine-readable format. P3P enabled browsers can "read" this snapshot automatically and compare it to the consumer's own set of privacy preferences. P3P enhances user control by putting privacy policies where users can find them, in a form users can understand, and, most importantly, enables users to act on what they see. (source: World Wide Web Consortium).

On some browsers (most notably, Internet Explorer 6.0) one of the most important browser settings controlled by P3P is whether cookies are accepted or rejected. In most cases, unless cookies are completely disabled, first party cookies will be accepted. For third party cookies to be accepted, the appropriate P3P header must be in place on all elements of your site (including the DCS). More information about configuration of P3P settings can be found in the DCS Administration Guide.

DCS allows different privacy policies to be applied to different parts of your site(s) based on the DCS JavaScript tag.

Cookies

The WebTrends Data Collection Server can be configured to serve cookies for use in sessionizing and in determining new vs. returning visitors. The cookie served will typically be in a 3rd party domain, as it is often used across multiple domains in an organization.

Some organizations will want to make use of a cookie that is already in place and other organizations do not allow 3rd party cookies and/or don't use persistent cookies or do not use cookies at all. DCS can be configured to accommodate any of these policies, applying cookies in different ways, or not using cookies at all.

If no cookie used, WebTrends solutions can sessionize based on IP address (unreliable) or URL session parameter (if used).

Bandwidth Requirements for DCS JavaScript

The Data Collection Server JavaScript does add some weight to pages, and the additional HTTP request made to the host to collect information does add overhead to serving the page. Fortunately, that amount of incremental overhead is minimal—the equivalent of adding a minor image to your page. In fact, the JavaScript tag is a mere 1K, which accounts for a less than one percent increase in size for most web pages. The JavaScript makes an HTTP GET request to the DCS for a 1 X 1 pixel transparent image, a request that requires roughly 400 bytes. That request, along with handshakes to connect with the DCS, adds a small amount of overhead, but typically, each view of a page containing the JavaScript requires less than 2K in incremental bandwidth.

A second reasonable question is whether the additional 1 X 1 pixel image and additional call to the DCS adds time to loading the page. The answer to this question varies widely depending on the proximity of the DCS to the client and the size of the pipe through which the web servers and DCS operate. In general, the total time added to the loading of a page is measured in fractions of a second, and should not be visible to the client. This is particularly true since the latest browsers can perform multiple tasks simultaneously, so it could continue to load other page elements while retrieving the DCS image from the DCS.

Inserting the JavaScript Tracking Code into the Page

For DCS to collect web traffic data for your web server, you must first modify a snippet of JavaScript so that it points to the DCS. You then need to include this code in any web pages from which you want to capture web traffic data. The unmodified JavaScript contains two references to "localhost", one for JavaScript-enabled web browsers, and one for web browsers that do not have JavaScript enabled. Both references to "localhost" should be replaced with the domain name for DCS.

For example, if your DCS was located at www.visitorstats.com, change the following code from:

```
SRC="http://localhost/dcs.gif?'+P+'"
     to
SRC="http://www.visitorstats.com/dcs.gif?'+P+'"
```

Within any web page that you plan to use DCS to capture web traffic information, insert the JavaScript anywhere between the <html> and </html> tags using NotePad or another text or web page editor that does not add extraneous html code to the page. It is recommended that the JavaScript be placed near the top of the web page; that way, if a user cancels before the entire page downloads, the page hit will still be sent to the DCS

The obvious question here is "Do I have to add the code to EVERY page on my site?" The answer is "Yes". The good news, however, is that there are many techniques that can be used to simplify this process, such as adding the code to a page template or common page element, such as a footer file. Very few sites would require page-by-page installation, and the average time required to implement this to most sites will be measured in hours, not days.

Passing Data From the Page via the JavaScript

The basic tag collects a standard set of valuable information about your web site; however, if you wish to capture additional information, you can. The general rule is that any component of the page can be captured by the DCS JavaScript and become available for reporting. To capture additional information, variables must be added to the JavaScript and defined to obtain the appropriate information. The information is then stored in the DCS as a parameter of the page's URL, and can later be used directly in WebTrends Reporting Center or stored in a warehouse for additional analysis. See the Administrator's Guide for WebTrends Data Collection Server for more details on this type of analysis.

```
<SCRIPT LANGUAGE="JavaScript">
<!--
P += "&dcsdat="+escape(sCurrent);

// Add your custom parameters here
var vProd = "Blue Shoes";
P += "&prod="+escape(vProd);
// End of customize code

document.write('<IMG BORDER="0" NAME="DCSIMG" WIDTH="1" HEIGHT="1"
SRC="http://localhost/dcs.gif?'+P+'">');
//-->
</SCRIPT>
```

The resulting URL would appear with the main body of the URL, plus "&prod=Blue Shoes".

Configuring DCS for Reporting Center

WebTrends Reporting Center 5.0 is designed to recognize certain parameters passed in the URL, and to "auto-configure" itself to report on these parameters. The variables in the JavaScript need to be passed in a specific way, but a large list of these parameters exists:

- Content Group
  - Sub Content Group
- Marketing Campaign
  - Name
  - Type
  - Cost
  - Start Time
  - End Time
- Ad View
- Ad Click
- Server Name
- Step of Interest for Scenario Analysis
- Page title
- Product name
- Product category
- Order type
- Units
- Subtotal
- Time zone
- User language
- Color depth
- Screen resolution
- Java enabled
- JavaScript enabled
- JavaScript version
- Cookie enabled
- Shopping carts
  - View
  - Add
  - Checkout
- First visit referrer
- First visit campaign
- First visit entry page
- First visit time
- Total visit count
- Visitor's overall $$ spent

DCS as a Hosted Service

WebTrends provides Data Collection Server functionality as a hosted service. With hosted DCS, a customer can receive aggregate web data daily for on-site analysis with WebTrends Reporting Center or WebTrends Intelligence Suite. The web data collected is placed on an FTP server daily.

When hosted, Data Collection Server is flexible enough to provide each individual customer with customized P3P and cookie settings. For P3P, WebTrends can apply a P3P policy that matches that used by your organization. With respect to cookies, an individual customer can choose to use cookie or to not use cookies at all. When cookies are not used, WebTrends Reporting Center will use an alternate means for visitor sessionization.

Differences Between Standard Log Files and DCS Files

In general, the output of the DCS server is similar to that of a standard web server. However, the method the DCS uses to collect its activity data results in a few, minor differences between a DCS-generated log file and the log files found on the originating content web server. Because the DCS requires the JavaScript code to be on any page to track it, pages such as error pages and redirects are not tracked, unless the code has been included in the specific page. In addition, if a page-load is interrupted before the browser gets to the JavaScript, there will be no record of a hit because the JavaScript never executes.

Error Pages

Error pages are captured in web server log files, but only by DCS if the error pages contain the JavaScript.

Non-JavaScript Enabled Browser

- Server-side log files contain the original hit, including the URL of the page being viewed.
- DCS log files contain a hit to /nojavascript, which will not log the name of the URL being viewed, as the variable cannot be fully defined without JavaScript enabled.

Redirects

- Server-side web server logs include all redirects.
- If the redirects do not include any JavaScript, they won't be captured in the DCS log file.

Interrupted Page Loading

- Server-side web server logs include hits for all pages served.
- DCS log files may not include a hit, if the page-load is interrupted. This depends on whether or not the JavaScript runs prior to the load interruption.

Crawlers and Spiders

- Server-side web server logs include hits from crawlers and spiders.
- If the crawler and spider do not execute the JavaScript, the DCS log file will not include the hit.

Non-Instrumented Content

- Server-side log files contain all hits to the web server.
- DCS log files only contain hits to pages that are DCS-instrumented.

Downloads and non-HTML Pages

- Server-side log files can include all downloads and non-HTML files (.html, .htm, .asp, etc.).
- With DCS, only viewable HTML pages can be captured (.html, .htm, .asp, etc.) without some custom configuration.

Proxy Servers and Page Caching

- With server-side log files, pages cached by proxy servers may not be captured.
- With DCS, all page views are captured.

Customized Parameters

- Server-side log files contain the original hit.
- DCS log files can be extended to contain advanced information in the form of custom parameters.

Collect Web Site Activity Data for All Your Sites and Domains

The Data Collection Server overcomes the barriers of getting web site activity information that many of today's web sites or families of web sites present. Through the execution of the Data Collection Server's JavaScript code, organizations with these barriers are able to collect valuable web site data for immediate use in WebTrends Reporting Center reports or for storage in a warehouse for later, more in-depth analysis with WebTrends Intelligence Suite. In addition, users of WebTrends Live, the fully-hosted e-service for web analysis and reporting can also use the DCS to store web activity data for more in-depth analysis.

By combining the information collected by the DCS with extensive analysis and reporting capabilities provided by other WebTrends solutions, your organization will be armed with the information required to improve web site effectiveness and return on investment.

WP10207DCS MP 0702